2002-05

Normative Design of Organizations - Part II: Organizational Structure

Levchuk, Georgiy M.

IEEE

IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, Vol. 32,

Normative Design of Organizations—Part II: Organizational Structure

Georgiy M. Levchuk, Yuri N. Levchuk, Jie Luo, Krishna R. Pattipati, *Fellow, IEEE*, and David L. Kleinman, *Fellow, IEEE*

***Abstract*—This paper presents a multiobjective structural optimization process of designing an organization to execute a specific mission. We provide mathematical formulations for optimization problems arising in Phases II and III of our organizational design process (Phase I was presented in Part I of this paper [56]) and polynomial algorithms to solve the corresponding problems. Our organizational design methodology applies specific optimization techniques at different phases of the design, efficiently matching the structure of a mission (in particular, the one defined by the courses of action obtained from mission planning) to that of an organization. It allows an analyst to obtain an acceptable tradeoff among multiple mission and design objectives, as well as between computational complexity and solution efficiency (desired degree of suboptimality).**

***Index Terms*—Clustering, organization structure, organizational design, organizational hierarchy, scheduling.**

I. Introduction

The OPTIMAL organizational design problem is one of finding both the optimal organizational structure (e.g., decision hierarchy, allocation of resources and functions to humans, communication structure, etc.) and strategy (allocation of tasks to decision-makers (DMs), scheduling task execution, detailing decision policies, etc.) that allow the organization to achieve superior performance, while conducting a specific mission [27]. Over the years, research in organizational decision-making has demonstrated that there exists a strong functional dependency between the specific structure of a mission environment and the concomitant optimal organizational design. Subsequently, it has been concluded that the optimality of an organizational design ultimately depends on the actual mission parameters (and organizational constraints). This premise led to the application of systems engineering techniques to the design of human teams. It advocates the use of normative algorithms for optimizing human team performance [24]–[29], [37], [38], [56]. This paper presents formulations and solution approaches for Phases II and III of our organizational design process (outlined in Part I of this paper [56]).

Manuscript received October 1, 2000; revised May 5, 2001. This work was supported by the Office of Naval Research under Contracts N00014-93-1-0793, N00014-98-1-0465 and N00014-00-1-0101. This paper was recommended by Associate Editor S. Ghosh.

G. M. Levchuk, J. Luo, and K. R. Pattipati are with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269 USA (e-mail: krishna@engr.uconn.edu).

D. L. Kleinman is with the Naval Post Graduate School, Monterrey, CA 93943 USA (e-mail: kleinman@nps.navy.mil).

Y. N. Levchuk is with Aptima Inc., Woburn, MA 01801 USA (e-mail: levchuk@aptima.com).

Digital Object Identifier 10.1109/TSMCA.2002.802820

A. Related Research

Over the past 15 years, research interest in teams and team performance has noticeably increased, spanning industrial and organizational psychology, operations research, business management, and decision-making in command-and-control. Many researchers have studied the interplay among the task environment, the team organization, and the team performance.

Addressing "the rapid expansion in the dimensions and complexity of contemporary team missions" [38], different types of task environments and the concomitant distributed organizations (e.g., joint task force organizations, flight crew of a commercial airline, collaborative software development teams, medical teams, research and development teams, etc.) have been studied, defining a variety of task and team variables relevant to team performance. For example, in studies of how emergency medical teams interact to resuscitate trauma patients [33], [52], the research found that team domain of trauma patient resuscitation embodies high risk, severe time pressure, high task complexity, extremely high levels of individual expertise, and highly distributed expertise from multiple specialists, including trauma surgeons and anesthesiologists. The team task also involves very high levels of uncertainty, including uncertainty about the nature and extent of the injury, the patient's prior medical history, the working status of the patient monitors (which may produce misleading readings), the effects of treatment, and the availability of other team members.

Studying the dimensions along which teams can be "distributed" (e.g., knowledge, expertise, information, resources, responsibility, authority, goals, etc.) underscored the complex nature of human interrelationships and compelled organizational researchers to extensively study organizational hierarchies (see, for example, [8], [16], [43]). It has been argued that, in hierarchically structured organizations, goal planning and strategy formulation occur typically in the upper levels [5]. The formal (centralized) organizations have explicit hierarchical structures, and they are efficient in task assignment and processing due to specialization of work and differentiation of roles. On the other hand, as information processing systems, hierarchies tend to filter the circulated information according to locally assessed goals, and, as the uncertainty increases, tendency to absorb information results in deterioration of organization's performance. Simon [44] argued that informal (decentralized) organizations also are hierarchically structured. He also discovered [13] that small groups within a team that are allowed unlimited choice of communication channels tend to centralize their communication flows into a hierarchical structure, thus supporting the claim that informal organizations will naturally evolve into a hierarchical structure.

Various mathematical measures of organizational design have been suggested in the literature to categorize teams (along multiple dimensions) and thus to enable the selection of appropriate performance improvement methods. Known measures of organizational design typically focus on either organizational structure (providing the information on who communicates with whom, or who directs/commands whom) or the task decomposition scheme (who has access to what resources, new data, and has responsibility for what portion/aspect of a task). Krachhardt in [5] developed several measures of organizational design from a graph theoretical perspective and argued about their relevance to performance. Mackenzie in [33] defined process indicators to demonstrate that, in certain cases, a high degree of hierarchy will enhance the effectiveness and efficiency of an organization.

Many attempts have been made to identify the performance and process measures most appropriate to a specific team domain (see, for example, [6], [14], [15], [20], [21], [48], [53], [54]). In general, however, there is little consensus on what constitutes organizational performance, and there is no universally best set of performance measures. As was shown in [4], whether an organization is said to perform well depends on the constraints placed on the performance measures and on organizational objectives. Performance has been viewed from a variety of perspectives, such as productivity [2], profitability [23], and reliability [40]. Although these measures may indicate what these organizations are doing, they do not always necessarily suggest how well they are doing it. Lin [5] gives a systematic evaluation of various performance criteria contrasting existing measures of organizational performance against each other and conducting simulation experiments to explore various aspects of organizations. The performance characteristics of simulated organizations were shown to be comparable (under certain conditions) to the performance characteristics observed in the real world [31].

The vast majority of research work addressing the improvement of team performance is heuristic in nature and deals with somewhat isolated aspects of a team (e.g., training, improving the lay-out of information acquisition systems, team selection, etc. [1], [9], [10], [16], [33]–[36], [42], [43], [45], [49], [50], [55]). Much fewer examples (e.g., [30], [38], [40], [41]) are known to actually address analytic methods to manage and improve team performance. In this paper, together with its companion paper [56], we focus on specific organizational objectives and constraints, and provide a theoretical framework for their use in model-based organizational design problem. Our optimized team structures exhibit superior performance with regard to specified organizational objectives.

B. Organization of the Paper

The paper is organized as follows. Section II presents an overview of our 3-phase design process. Section III defines the optimization problem arising in Phase II, and provides algorithms to solve it. Section IV presents the formulation of structural optimization problem (Phase III), and discusses the objective functions and the corresponding algorithms used to optimize organizational hierarchy. Section V provides a discussion on algorithm performance and effects of optimization parameters and objectives on the organizational structure. The paper concludes with a summary and future extensions in Section VI.

II. 3-Phase Organizational Design Process

When modeling a complex mission and designing the corresponding organization, the variety of mission dimensions (e.g., functional interdependencies, geographical layout, information processing, etc.), together with the required level of model granularity (e.g., mission task and organizational unit decompositions), determines the complexity of the design process. Our mission modeling and organizational design methodology allow one to overcome the computational complexity by synthesizing an organizational structure via an iterative solution of a sequence of three smaller and well-defined optimization problems [25], [56]. The three phases of our design process solve three distinct optimization subproblems.

Phase I (Scheduling Phase): The first phase of our design process determines the task-platform allocation and task sequencing that optimize mission objectives (e.g., mission completion time, accuracy, workload, resource utilization, platform coordination, etc.), taking into account task precedence constraints and synchronization delays, task resource requirements, resource capabilities, as well as geographical and other task transition constraints. The generated task-platform allocation schedule specifies the workload of each resource. In addition, for every mission task, the first phase of the algorithm delineates a set of nonredundant resource packages capable of jointly processing a task. This information is later used for iterative refinement of the design, and, if necessary, for on-line strategy adjustments.

Phase II (Clustering Phase): In this phase, we combine platforms into nonintersecting groups, to match the operational expertise and workload threshold constraints on available DMs, and assign each group to an individual DM to define the DM-resource allocation. Thus, the second phase delineates the DM-platform-task allocation schedule and, consequently, the individual operational workload of each DM.

Phase III (Structural Optimization Phase): Finally, Phase III completes the design by specifying a communication structure and a decision hierarchy to optimize the responsibility distribution and inter-DM control coordination, as well as to balance the control workload among DMs according to their expertise constraints.

In this paper, we present mathematical formulations of clustering and network-configuration problems (arising in Phases II and III of our organizational design process) and describe polynomial algorithms to solve these problems. For an overview of our organizational design process, its mission-planning (scheduling) phase, and related research, see [56].

III. Phase II: DM-Resource Allocation

The second phase of our design process combines resources into nonoverlapping groups to match the operational expertise and workload threshold constraints of available DMs. It assigns each group to an individual DM to define the DM-resource allocation and a consequent DM-platform-task schedule. The latter also specifies the (dynamic) individual operational workload of each DM.

Since the decision-making and operational capabilities of a human are limited, the distribution of *information*, *resources*, and *activities* among DMs must be set up to achieve timely mission processing while efficiently utilizing each DM. The total load is generally partitioned among DMs by decomposing a mission into tasks and assigning these tasks to individual DMs who are responsible for their planing and execution. Moreover, an overlap in task processing (wherein two or more DMs share responsibility for a given function/task while each possessing the capability to individually process the task) gives the team a degree of freedom to adapt to uneven demand by redistributing the task processing load. The critical issues in team *task processing* are: *what* should be done, *who* should do what, and *when*.

In general, DMs are provided with limited resources with which to accomplish their objectives. The distribution of these resources among DMs and the assignment of these resources that enables task processing are among key elements defining an organizational design. Team members must dynamically coordinate their resources to process their individual tasks, while assuring that team performance goals are met. The critical issues in team *resource allocation* are: *who* should *own which* resource, *who* should use which resource to *do what*, and *when*.

The allocation of information/resources/tasks to DMs is equivalent to first grouping the corresponding entities and then assigning each group to a different DM. The basis for such a grouping can be obtained by a cluster analysis of the corresponding objects or entities. Objects (e.g., platforms, resources, tasks) that are described by their relationship to other objects can be classified according to their perceived similarities. Clustering then can be used to partition the set of objects into distinct, mutually exclusive subsets (clusters) of *similar* objects to achieve the prescribed relationships among cluster groups.

Specifically, to allocate resources and tasks to DMs, our organizational design process makes use of the task-platform assignment results, obtained in its Phase I (described in [56]), as follows. The platforms are grouped into disjoint clusters according to their task assignments, and these platform clusters are then allocated to different DMs who inherit the corresponding task assignments. The objective of platform clustering is to minimize the resultant DM workload—a weighted sum of external DM–DM coordination and internal platform coordination load of a DM, formally defined below.

A. Problem Definition

The following assignment data from Phase I are used to define the problem:

$$w_{ij} = \begin{cases} 1, & \text{if platform } P_j \text{ is assigned task } T_i \\ 0, & \text{otherwise} \end{cases}$$

$$K = \text{number of platforms}$$

$$N = \text{number of tasks}$$

$$D = \text{number of DMs.}$$

We also define:

$$x_{mj} = \begin{cases} 1, & \text{if } DM_m \text{ is allocated to platform } P_j \\ 0, & \text{otherwise.} \end{cases}$$

A platform-task assignment specifies the necessary interaction among platforms when processing a task. This interaction necessitates coordination among DMs, assigned to these platforms as information/decision/action carriers. Specifically, to model coordination-related overhead in an organization, we define two types of coordination: *1) internal* and *2) external*. Internal coordination accounts for the need to coordinate among platforms assigned to the same DM. External coordination is the inter-DM coordination that results from a multi-DM task assignment.

The formal definitions used for internal and external coordination are as follows.

Definition 1: A *signature vector* of a DM DM_m is a DM-task assignment vector

$$[u_{m1}, \ldots, u_{mN}]$$

where

$$u_{mi} = \begin{cases} 1, & \text{if } DM_m \text{ is assigned to task } T_i \\ 0, & \text{otherwise} \end{cases}$$
$$= \begin{cases} 1, & \text{if there exists a platform } P_j \\ & \quad \text{such that } x_{mj} = 1,\, w_{ij} = 1 \\ 0, & \text{otherwise.} \end{cases}$$

In a similar vein, the signature vectors are defined for any group of platforms (not necessarily associated with a DM).

Definition 2: The *internal coordination* of a DM_m is equal to the number of platforms assigned to this DM

$$\boldsymbol{I}(m) = \sum_{j=1}^{K} x_{mj}. \tag{1}$$

Definition 3: A *direct DM–DM coordination* between two DMs DM_m and DM_n is equal to the number of tasks simultaneously processed by these DMs

$$\boldsymbol{D}(m, n) = \sum_{i=1}^{N} u_{mi}u_{ni} = \sum_{i=1}^{N} \min(u_{mi}, u_{ni}). \tag{2}$$

Definition 4: The *external coordination* of a DM_m is the sum of its direct coordinations with other DMs

$$\boldsymbol{E}(m) = \sum_{\substack{n=1 \\ n\neq m}}^{D} \boldsymbol{D}(m, n). \tag{3}$$

Definition 5: Coordination Workload of a DM_m is a weighted sum of internal and external coordination of this DM

$$\boldsymbol{CW}(m) = W^I \cdot \boldsymbol{I}(m) + W^E \cdot \boldsymbol{E}(m). \tag{4}$$

Weights for internal (W^I) and external (W^E) coordination specify their impact on the corresponding aggregated DM workload.

Given the data from Phase I, platforms are clustered into groups to be assigned to DMs. The objective is to minimize the maximal DM coordination workload associated with DM-platform-task assignment.

Example: Experiment With DDD-III Simulator: For our example from an experiment with the DDD-III simulator (described in detail in [56]), platform-task allocation obtained in the scheduling phase (Phase I) via pairwise exchange (PWE) algorithm is used. For DM-platform assignment in Fig. 1 (which is optimal for this example), internal coordination is

$$\underline{\boldsymbol{I}} = [\boldsymbol{I}(1), \ldots, \boldsymbol{I}(5)] = [5\ \ 2\ \ 4\ \ 5\ \ 4]$$

direct DM–DM coordination is

$$[\boldsymbol{D}(i, j)] = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}.$$

External DM coordination is

$$\underline{\boldsymbol{E}} = [\boldsymbol{E}(1), \ldots, \boldsymbol{E}(5)] = [2\ \ 2\ \ 1\ \ 2\ \ 3]$$

and DM coordination workload for $W^I = W^E = 1$ is

$$\underline{\boldsymbol{CW}} = [\boldsymbol{CW}(1), \ldots, \boldsymbol{CW}(5)] = \underline{\boldsymbol{I}} + \underline{\boldsymbol{E}} = [7\ \ 4\ \ 5\ \ 7\ \ 7].$$

B. Related Research

Cluster analysis and the corresponding grouping of objects are generally used to achieve two main objectives. The first objective is to *maximize* the distance (dissimilarity) between clusters. The second objective is to *minimize* the dissimilarity among the objects in the same cluster (for each respective cluster). The first objective can be achieved by using *single-link* methods to find clusters with minimal path lengths among all objects in the cluster, while the second objective can be achieved by using *complete-link* methods to find clusters with minimum diameter [18], [19]. Other algorithms have been developed for combinations of these objectives, such as UPGMA (group average), WPGMA (weighted average), UPGMC (unweighted centroid), WPGMC (weighted centroid) (see [46]), and Ward's method to minimize square-error [51]. The generalization of the above methods was presented in [22].

In this paper, we deal with *agglomerative* hierarchical clustering. This procedure starts with disjoint clustering, which places each of the objects into an individual cluster. The process is repeated to form a sequence of nested groups in which the number of clusters decreases as the sequence progresses. For a review of clustering algorithms, see [11] and [18].

All of the algorithms employed for cluster analysis assume that the distance (or dissimilarity) between objects is easily obtained and updated. In our case, we not only need to consider distances between objects, but also must take into account the number of objects in the cluster (number of platforms in the cluster constitutes the internal coordination of a DM assigned to operate these platforms). Therefore, existing approaches need to be modified to obtain algorithms suited for our problem.

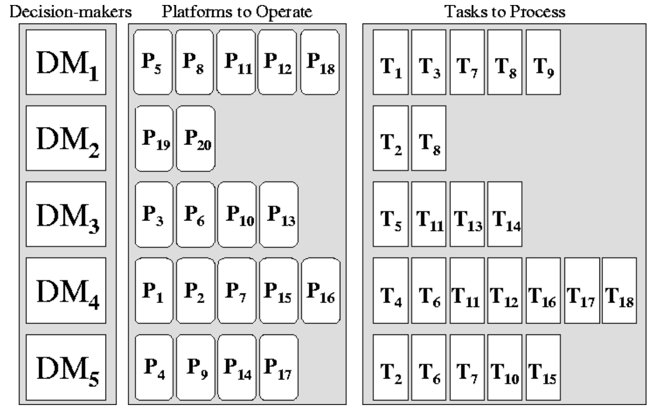

Fig. 1. Optimal DM-platform-task allocation for example with DDD-III simulator $W^I = W^E = 1$.

C. Mathematical Formulation of the Clustering Problem

A fundamental question underlying a distributed organizational design—"*Who should do which part of the mission*?"—implies that the mission must be *decomposable* into a set of *entities*. These entities are generally referred to as *tasks*.

The following additional variables are used to define the clustering problem associated with the Phase II of our 3-phase organizational design process:

$$y_{nmi} = \begin{cases} 1, & \text{if } DM_n \text{ and } DM_m \text{ coordinate over task } T_i \\ 0, & \text{otherwise} \end{cases}$$
$$C_W = \text{maximal weighted coordination workload.}$$

We note that $y_{nmi} = u_{ni} \cdot u_{mi} = \min(u_{ni}, u_{mi})$.

Following [29], the problem assumes the form of a binary (0–1) programming problem

$$\min C_W$$
$$\begin{cases} \sum_{m=1}^{D} x_{mj} = 1, & j = 1, \ldots, K \\ y_{nmi} \geq w_{ji} \cdot x_{mj}, & m, n = 1, \ldots, D;\ i = 1, \ldots, N;\ j = 1, \ldots, K \\ y_{nmi} \geq w_{ji} \cdot x_{nj}, & m, n = 1, \ldots, D;\ i = 1, \ldots, N;\ j = 1, \ldots, K \\ C_W \geq W^I \cdot \sum_{j=1}^{K} x_{nj} + W^E \cdot \sum_{m=1, m\neq n}^{D} \sum_{i=1}^{N} y_{nmi}, & n = 1, \ldots, D \\ & x_{nj}, y_{nmi} \in \{0, 1\} \end{cases}. \tag{5}$$

This problem is NP-hard [12]. Near-optimal heuristic clustering algorithms exist that are specifically customized for this problem.

D. Cluster Merging

Suppose there exist M clusters (groups of platforms) $\{G_m, m = 1, \ldots, M\}$ determined by the assignment variables x_{mj} with the corresponding signature vectors $[u_{m1}, \ldots, u_{mN}]$. Then, the inter-group direct coordination and external group coordination can be found as follows:

$$\boldsymbol{D}(m, n) = \sum_{i=1}^{N} u_{mi}u_{ni} = \sum_{i=1}^{N} \min(u_{mi}, u_{ni}) \tag{6}$$

$$\boldsymbol{E}(m) = \sum_{\substack{n=1\\ n\neq m}}^{M} \boldsymbol{D}(m, n). \tag{7}$$

The size of each group is $\boldsymbol{I}(m) = \sum_{j=1}^{K} x_{mj}$. When two groups r and s are merged together into a new group φ, the signature vector for the new group is

$$[u_{\varphi 1}, \ldots, u_{\varphi N}] = [\max(u_{r1}, u_{s1}), \ldots, \max(u_{rN}, u_{sN})].$$

Direct coordination and cluster sizes are updated accordingly

$$\begin{aligned}\boldsymbol{D}(m, \varphi) &\leftarrow \boldsymbol{D}(m, r) + \boldsymbol{D}(m, s) - \sum_{i=1}^{N} \min(u_{ri}, u_{si}) \cdot u_{mi}, \\ &\qquad m \neq r, s \\ \boldsymbol{I}(\varphi) &\leftarrow \boldsymbol{I}(r) + \boldsymbol{I}(s).\end{aligned}$$

Then, the external coordination of the new clusters is

$$\begin{aligned}\boldsymbol{E}(\varphi) &= \sum_{i=1}^{N} \max(u_{ri}, u_{si}) \cdot \sum_{m=1, m\neq r, s}^{M} u_{mi} \\ \boldsymbol{E}(m) &= \boldsymbol{E}(m) - (\boldsymbol{D}(m, r) + \boldsymbol{D}(m, s)) + \boldsymbol{D}(m, \varphi) \\ &\qquad m \neq r, s.\end{aligned} \tag{8}$$

Clusters that are unchanged by the merger have nonincreasing external coordination, while maintaining the same internal coordination. Therefore, their workload does not increase under cluster merging. A rule for selecting the clusters to be merged influences the cluster workload. In the subsections E and F, we exploit this behavior to propose two algorithms for cluster selection.

Example (continued): For a cluster merging depicted in Fig. 2, DM–DM direct coordination matrix is updated as follows:

(A) row 5 and column 5 are deleted;

(B) row 4 and column 4 are updated according to (6).

For the grouping in Fig. 2, we update $[\boldsymbol{D}(i, j)]$ as follows:

$$\begin{bmatrix} 0 & 1 & 0 & 0 & \mathbf{0} & 1 \\ 1 & 0 & 0 & 0 & \mathbf{0} & 1 \\ 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & 1 \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ 1 & 1 & 0 & 1 & \mathbf{0} & 0 \end{bmatrix}$$

$$\xrightarrow{(\boldsymbol{A})} \begin{bmatrix} 0 & 1 & 0 & \mathbf{0} & 1 \\ 1 & 0 & 0 & \mathbf{0} & 1 \\ 0 & 0 & 0 & \mathbf{0} & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ 1 & 1 & 0 & \mathbf{1} & 0 \end{bmatrix} \xrightarrow{(\boldsymbol{B})} \begin{bmatrix} 0 & 1 & 0 & \mathbf{0} & 1 \\ 1 & 0 & 0 & \mathbf{0} & 1 \\ 0 & 0 & 0 & \mathbf{1} & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} \\ 1 & 1 & 0 & \mathbf{1} & 0 \end{bmatrix}$$

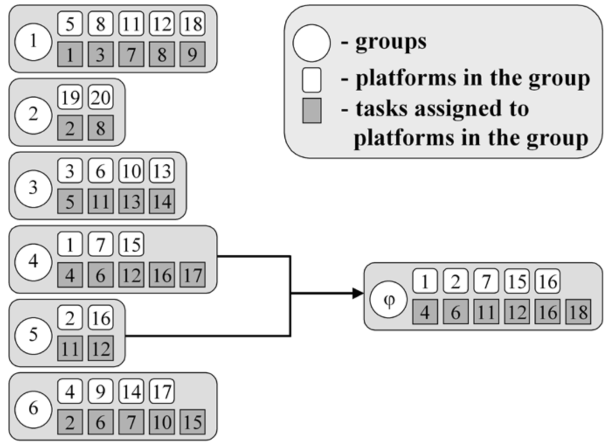

Fig. 2. Cluster merging for example with DDD-III simulator $W^I = W^E = 1$.

where

$$\begin{aligned}&[\boldsymbol{D}(4, 1), \ldots, \boldsymbol{D}(4, 5)] \\ &\quad\leftarrow \left[\boldsymbol{D}^{old}(4, 1), \boldsymbol{D}^{old}(4, 2), \boldsymbol{D}^{old}(4, 3), 0, \boldsymbol{D}^{old}(4, 6)\right] \\ &\qquad + \left[\boldsymbol{D}^{old}(5, 1), \boldsymbol{D}^{old}(5, 2), \boldsymbol{D}^{old}(5, 3), 0, \boldsymbol{D}^{old}(5, 6)\right] \\ &\qquad - [u^{old}_{1,12}, u^{old}_{2,12}, u^{old}_{3,12}, 0, u^{old}_{6,12}] \\ &\quad= [0, 0, 0, 0, 1] + [0, 0, 1, 0, 0] - [0, 0, 0, 0, 0] \\ &\quad= [0, 0, 1, 0, 1].\end{aligned}$$

Internal coordination is updated as

$$\begin{aligned}&\underline{\boldsymbol{I}} = [\boldsymbol{I}(1), \ldots, \boldsymbol{I}(5)] \\ &\quad\leftarrow \left[\boldsymbol{I}^{old}(1), \boldsymbol{I}^{old}(2), \boldsymbol{I}^{old}(3), \boldsymbol{I}^{old}(4) + \boldsymbol{I}^{old}(5), \boldsymbol{I}^{old}(6)\right] \\ &\quad= [5, 2, 4, 3+2, 4] = [5, 2, 4, 5, 4]\end{aligned}$$

and the new external coordination is found via

$$\begin{aligned}\boldsymbol{E}(\varphi) &= \sum_{i=1}^{18} \max(u_{ri}, u_{si}) \cdot \sum_{m=1, m\neq 4, 5}^{18} u_{mi} \\ &= \sum_{i\in\{4, 6, 11, 12, 16, 17, 18\}} \sum_{m=1, m\neq 4, 5}^{18} u_{mi} \\ &= 0+1+1+0+0+0+0 = 2, \\ \underline{\boldsymbol{E}} &= [\boldsymbol{E}(1), \ldots, \boldsymbol{E}(5)] \\ &\leftarrow \left[\boldsymbol{E}^{old}(1), \boldsymbol{E}^{old}(2), \boldsymbol{E}^{old}(3), \boldsymbol{E}(\varphi), \boldsymbol{E}^{old}(6)\right] \\ &= [2, 2, 1, 2, 3].\end{aligned}$$

E. Min-Dissimilarity Clustering

This algorithm employs minimum-dissimilarity clustering technique. We specify dissimilarities between clusters (groups of platforms) according to their coordination (derived from the corresponding platform assignment and task assignment data)

$$\boldsymbol{d}(m, n) = W^I \cdot (\boldsymbol{I}(m) + \boldsymbol{I}(n)) - W^E \cdot \boldsymbol{D}(m, n). \tag{9}$$

Two groups r and s with minimum dissimilarity $d(r, s)$ are merged together. Note that the coefficients $d(m, n)$ do not satisfy distance properties (such as triangle inequality, etc.). The groups of platforms are merged together to obtain a tradeoff between two objectives:

1) minimizing the new group size;
2) "removing" the largest direct coordination.

This tradeoff is derived from the correlation between the internal and external coordination weights (see Definition 5). The updates needed for the algorithm take $O(K + N(M-1))$ operations

$$\begin{aligned} &x_{\varphi j} \leftarrow \max(x_{rj}, x_{sj}) \\ &u_{\varphi i} \leftarrow \max(u_{ri}, u_{si}) \\ &I(\varphi) \leftarrow I(r) + I(s) \\ &\boldsymbol{d}(m, \varphi) \leftarrow W^I \cdot (\boldsymbol{I}(m) + \boldsymbol{I}(\varphi)) - W^E \cdot \sum_{i=1}^{N} u_{\varphi i} \cdot u_{mi}; \\ &\qquad m \neq r, s. \end{aligned}$$

We use a heap implementation [maintaining at each step a tree of $((M-1)(M-2))/2$ nodes] for dissimilarity values. At each step (merger) of the algorithm, the dissimilarity matrix update can be viewed as $(M-2)$ element updates (which corresponds to the sift-down operations in the heap) and $(M-1)$ delete operations (including finding and deleting the minimum element).

Therefore, the process of selecting the minimum element and updating dissimilarity values via heap takes

$$O\left(p \cdot (2M-3) \log_p \left(\frac{(M-1)(M-2)}{2}\right)\right)$$

operations, and the overall complexity of the algorithm is

$$O\left(\begin{array}{l} (K-D)\left(K + \dfrac{N}{2}(K+D-3)\right) \\ + p \log_p \left(\displaystyle\prod_{M=D+1}^{K} \left[\frac{(M-1)(M-2)}{2}\right]^{(2M-3)}\right) \end{array}\right)$$

where p is a coefficient equal to the largest number of children of any node in the heap. We can observe that in order to minimize the above concave function, the coefficient p must be chosen to be equal to either three or four.

Another variation of *min-dissimilarity* clustering algorithm, called *max-similarity*, calculates the proximity between clusters as

$$\boldsymbol{d}(m, n) = W^I (\boldsymbol{I}(m) + \boldsymbol{I}(n)) - W^E (\boldsymbol{D}(m, n) - \boldsymbol{Z}(m, n)) \tag{10}$$

where

$$\boldsymbol{Z}(m, n) = \sum_{i=1}^{N} \mathbf{1}(u_{ni} + u_{mi} = 1)$$

$$\mathbf{1}(A) = \begin{cases} 1, & \text{if } A = true \\ 0, & \text{otherwise.} \end{cases}$$

This algorithm tries to merge clusters having similar signature vectors. Algorithm complexity and coefficient updates are similar to *min-dissimilarity* algorithm.

Example (continued): For platform grouping shown in Fig. 2, dissimilarity matrix between clusters for *min-dissimilarity* algorithm is

$$[\boldsymbol{d}(i, j)] = \begin{bmatrix} - & 7 & 9 & 8 & 7 & 9 \\ 7 & - & 6 & 5 & 4 & 6 \\ 9 & 6 & - & 7 & 6 & 8 \\ 8 & 5 & 7 & - & 5 & 7 \\ 7 & 4 & 6 & 5 & - & 6 \\ 9 & 6 & 8 & 7 & 6 & - \end{bmatrix}$$

$$- \begin{bmatrix} - & 1 & 0 & 0 & 0 & 1 \\ 1 & - & 0 & 0 & 0 & 1 \\ 0 & 0 & - & 0 & 1 & 0 \\ 0 & 0 & 0 & - & 1 & 1 \\ 0 & 0 & 1 & 1 & - & 0 \\ 1 & 1 & 0 & 1 & 0 & - \end{bmatrix} = \begin{bmatrix} - & 6 & 9 & 8 & 7 & 8 \\ 6 & - & 6 & 5 & \mathbf{4} & 5 \\ 9 & 6 & - & 7 & 5 & 8 \\ 8 & 5 & 7 & - & \mathbf{4} & 6 \\ 7 & 4 & 5 & 4 & - & 6 \\ 8 & 5 & 8 & 6 & 6 & - \end{bmatrix}.$$

There are two possibilities: 1) merge groups 2 and 5; or 2) merge groups 4 and 5. For 2) (see Fig. 2), the dissimilarity matrix $[\boldsymbol{d}(i, j)]$ is updated similar to $[\boldsymbol{D}(i, j)]$

$$\begin{bmatrix} - & 6 & 9 & 8 & \mathbf{7} & 8 \\ 6 & - & 6 & 5 & \mathbf{4} & 5 \\ 9 & 6 & - & 7 & \mathbf{5} & 8 \\ 8 & 5 & 7 & - & \mathbf{4} & 6 \\ \mathbf{7} & \mathbf{4} & \mathbf{5} & \mathbf{4} & - & \mathbf{6} \\ 8 & 5 & 8 & 6 & \mathbf{6} & - \end{bmatrix}$$

$$\xrightarrow{(\boldsymbol{A})} \begin{bmatrix} - & 6 & 9 & \mathbf{8} & 8 \\ 6 & - & 6 & \mathbf{5} & 5 \\ 9 & 6 & - & \mathbf{7} & 8 \\ \mathbf{8} & \mathbf{5} & \mathbf{7} & - & \mathbf{6} \\ 8 & 5 & 8 & \mathbf{6} & - \end{bmatrix} \xrightarrow{(\boldsymbol{B})} \begin{bmatrix} - & 6 & 9 & 9 & 8 \\ 6 & - & 6 & 7 & 5 \\ 9 & 6 & - & 10 & 8 \\ 9 & 7 & 10 & - & 10 \\ 8 & 5 & 8 & 10 & - \end{bmatrix}$$

where

$$\begin{aligned} &[\boldsymbol{d}(4, 1), \ldots, \boldsymbol{d}(4, 5)] \\ &\quad \leftarrow [\boldsymbol{I}(1) + \boldsymbol{I}(4), \ldots, \boldsymbol{I}(5) + \boldsymbol{I}(4)] \\ &\qquad + [\boldsymbol{D}(4, 1), \ldots, \boldsymbol{D}(4, 5)] \\ &\quad = [4+5, 2+5, 4+5, -, 4+5] - [0, 0, 1, -, 1] \\ &\quad = [9, 7, 10, -, 10]. \end{aligned}$$

F. Best-Merge Clustering

This algorithm finds a merge of two groups of platforms that produces either the largest decrease or the smallest increase in the objective function of Phase II. At each step, the maximum workload of the new cluster produced by such a merger is found, and the merge with the lowest maximum workload is selected. If "ties" occur, a group with the smallest $d(r, s)$ is selected. When clusters r and s are merged, group workload is updated as

$$\begin{aligned} &\boldsymbol{CW}(m) \leftarrow \boldsymbol{CW}(m) - W^E \cdot \sum_{i=1}^{N} \min(u_{ri}, u_{si}) \cdot u_{mi}; \\ &\qquad m \neq r, s \\ &\boldsymbol{CW}(\varphi) \leftarrow \boldsymbol{CW}(r) + \boldsymbol{CW}(s) - W^E \\ &\qquad \cdot \left(2\boldsymbol{D}(r, s) + \sum_{\substack{m=1 \\ m \neq r, s}}^{M} \sum_{i=1}^{N} \min(u_{ri}, u_{si}) \cdot u_{mi}\right). \end{aligned}$$

For each possible cluster pair $\{n, k\}$ merge, we evaluate

$$\boldsymbol{CW}_{nk}(m) \leftarrow \boldsymbol{CW}(m) - W^E \cdot \boldsymbol{\Delta}(m, n, k); \qquad m \neq n, k$$

$$\boldsymbol{CW}_{nk}(\varphi) \leftarrow \boldsymbol{CW}(n) + \boldsymbol{CW}(k) - W^E \cdot \left(2\boldsymbol{D}(n, k) + \sum_{\substack{m=1 \\ m \neq n, k}}^{M} \boldsymbol{\Delta}(m, n, k) \right)$$

where

$$\boldsymbol{\Delta}(m, n, k) = \sum_{i=1}^{N} \min(u_{ni}, u_{ki}) \cdot u_{mi} = \sum_{i=1}^{N} u_{ni} \cdot u_{ki} \cdot u_{mi}.$$

The maximal workload is then found as

$$\boldsymbol{CW}\mathbf{max}_{nk} = \max\{\boldsymbol{CW}_{nk}(\varphi), \boldsymbol{CW}_{nk}(m);\, m \neq n, k\}.$$

Two clusters, r and s, are selected as follows:

$$(r, s) = \arg\min_{n, k}(\boldsymbol{CW}\mathbf{max}_{nk}).$$

When clusters are merged, the following parameters are updated:

$$x_{\varphi j} \leftarrow \max(x_{rj}, x_{sj})$$
$$u_{\varphi i} \leftarrow \max(u_{ri}, u_{si})$$
$$\boldsymbol{D}(m, \varphi) \leftarrow \boldsymbol{D}(m, r) + \boldsymbol{D}(m, s) - \boldsymbol{\Delta}(m, r, s)$$
$$\boldsymbol{\Delta}(\varphi, n, k) \leftarrow \sum_{i=1}^{N} \min(u_{ni}, u_{ki}) \cdot u_{\varphi i}$$
$$\boldsymbol{\Delta}(n, \varphi, k) = \boldsymbol{\Delta}(n, k, \varphi) = \boldsymbol{\Delta}(\varphi, n, k) \quad m, n, k \neq r, s.$$

Cluster pair selection requires $O((M-1)(M-2)^2)$ operations, and cluster parameter update needs $O(K + 2(M-2) + N(M-2)(M-3))$ operations. Therefore, the overall complexity is approximately

$$O\left(K^4 - D^4 + N(K^3 - D^3)\right).$$

Example (continued): For platform grouping shown in Fig. 2, we have

$$\boldsymbol{\Delta}(m, n, k) = 0, \qquad m, n, k \in \{1, \ldots, 6\}.$$

Therefore, the matrix $[\boldsymbol{CW}\mathbf{max}_{nk}]$ is shown at the bottom of the page (e.g., $\boldsymbol{CW}\mathbf{max}_{4,5} = \max[\boldsymbol{CW}(1), \boldsymbol{CW}(2), \boldsymbol{CW}(3), \boldsymbol{CW}(4) + \boldsymbol{CW}(5) - 2D(4, 5), \boldsymbol{CW}(6)] = \max[7, 4, 5, 5 + 4 - 2 \cdot 1, 7] = 7$).

Hence, there are two possibilities to obtain the least maximal coordination workload: 1) merge groups 3 and 5; or 2) merge groups 4 and 5 (see Fig. 2). The resulting maximal coordination workload obtained by Best-Merge method is seven units.

G. Hierarchical Clustering Algorithm

Initialization: Begin by assigning each platform to a distinct cluster with the *signature vectors*

$$G_m = [u_{m1}, \ldots, u_{mN}].$$

```
Step 1. Choose two clusters (under min-
  dissimilarity or Best Workload rule) and
  combine them into a single cluster. Find
  the signature vector for the new cluster
  and update the distance matrix.
Step 2. If number of clusters is equal to
  the number of available DMs, STOP. Oth-
  erwise, go to Step 1.
```

Example (continued): Platform-task allocation obtained in the scheduling phase (Phase I) via PWE algorithm is used. Fig. 3 shows DM coordination networks (DCNs) corresponding to three DM-platform assignments obtained by various algorithms with internal/external workload weights $W^I = W^E = 1$.

It was found that minimizing the internal workload increases DMs external workload, thereby generating dense and heavy coordination network among DMs.

IV. Phase III: Organizational Hierarchy

To avoid decision-making confusion associated with the distribution of control, organizations may impose a decision hierarchy (i.e., *superior–subordinate* or *supported–supporting relations*) on their team members. A hierarchy is a partial order relationship that can be viewed as a tree-type network among DM nodes (with "root" DM being the team leader). Oftentimes, a hierarchy induces a structure for decision cycles

$$\begin{bmatrix} - & \max[9,5,5,4,7] & \max[12,4,5,4,7] & \max[12,4,5,4,7] & \max[11,4,5,5,7] & \max[12,4,5,5,4] \\ \cdots & - & \max[7,9,5,4,7] & \max[7,9,5,4,7] & \max[7,8,5,5,7] & \max[7,9,5,5,4] \\ \cdots & \cdots & - & \max[7,4,10,4,7] & \max[7,4,7,5,7] & \max[7,4,12,5,4] \\ \cdots & \cdots & \cdots & - & \max[7,4,5,7,7] & \max[7,4,5,10,4] \\ \cdots & \cdots & \cdots & \cdots & - & \max[7,4,5,5,11] \\ \cdots & \cdots & \cdots & \cdots & \cdots & - \end{bmatrix}$$

$$= \begin{bmatrix} - & 9 & 12 & 12 & 11 & 12 \\ 9 & - & 9 & 9 & 8 & 9 \\ 12 & 9 & - & 10 & \mathbf{7} & 12 \\ 12 & 9 & 10 & - & \mathbf{7} & 10 \\ 11 & 8 & 7 & 7 & - & 11 \\ 12 & 9 & 12 & 10 & 11 & - \end{bmatrix}$$

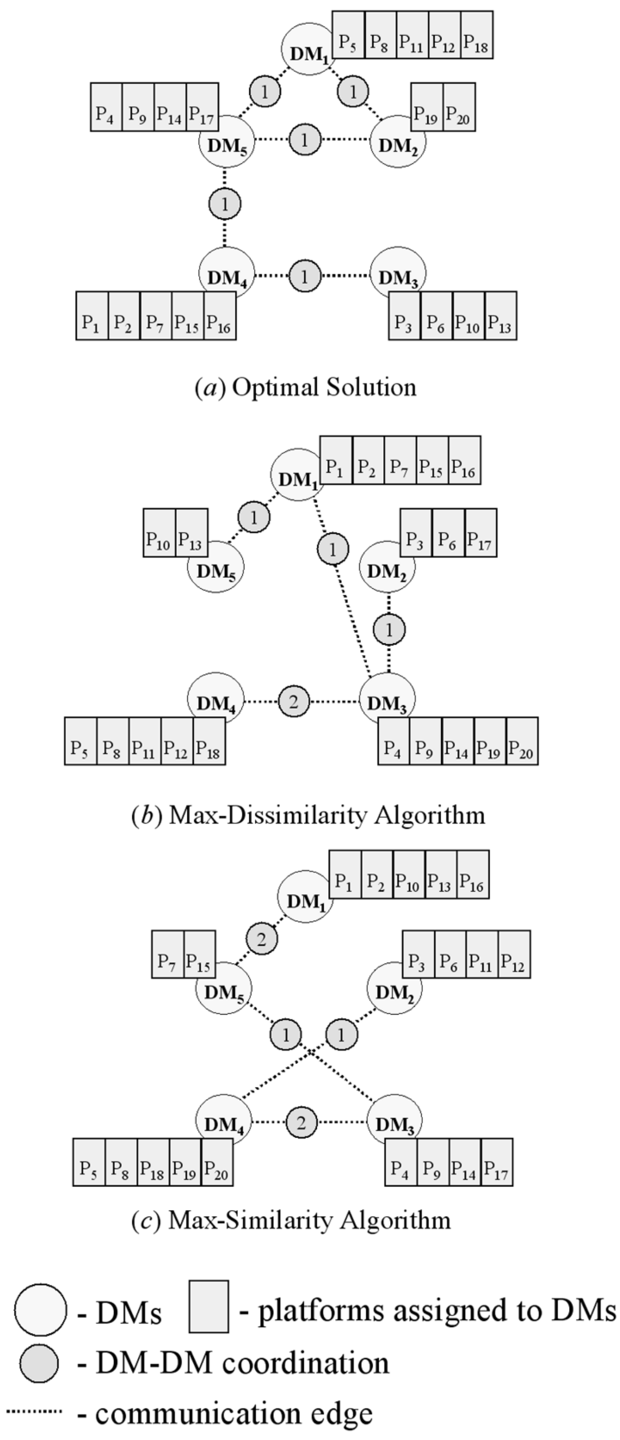

Fig. 3. Clustering results obtained by various algorithms for an experiment with the DDD-III simulator $W^I = W^E = 1$.

and information flows associated with inter-DM coordination in an organization. One of the goals in creating a specific hierarchy is to match the induced superior-subordinate DM relationships with the inter-DM coordination required to complete the mission. Different definitions of this matching lead to different formulations of the organizational hierarchy-design problem.

Phase III of our optimization process completes the organizational design by specifying: 1) a communication structure; and 2) a decision hierarchy (a directed tree spanning DM nodes) to optimize the *responsibility distribution* and *inter-DM control coordination*, as well as to balance the *control workload* among DMs according to their *expertise constraints*. The hierarchical structure of the organization ("who reports to whom") is uniquely determined by specifying the root node in the coordination tree of DM nodes. Different optimization objectives for the prospective organizational design (e.g., maximizing the *speed of command* by minimizing average decision cycles in the organization; minimizing the *management cost* associated with coordination overhead; etc.) prompt different rules for building a hierarchy and selecting its root. Some of the rules for root selection are as follows:

1) minimum tree depth;
2) DM with minimum workload;
3) DM with maximum coordination.

A. Problem Definition

In this section, we consider the situation when the coordination between any two DMs in the prospective organization requires the participation (e.g., approval) of all DMs involved in the corresponding superior–subordinate path spanning two coordinating DMs (e.g., when passing certain information is permitted only via hierarchy tree links, when each DM can communicate exclusively with his immediate superior/subordinate DMs, etc.). Such a spanning path is unique for any two DMs in a given DM hierarchy. The associated *coordination overhead* adds the extra load to each DM involved in the decision cycle. We model such an overhead by introducing *indirect additional coordination*. The *external organizational workload* is then defined as the sum of direct (one-to-one) and indirect (through an intermediary) coordination. The impact of each such coordination can be defined appropriately by introducing weighting coefficients.

We use the following definitions to formulate the problem.

Definition 6: DM coordination network (DCN) is a complete undirected graph with nodes $\{DM_1, \ldots, DM_D\}$ representing the DMs and edges with weights c_{mn} between nodes equal to the amount of necessary *direct coordination* between DMs (obtained in Phase II from DM-platform-task assignment)

$$c_{mn} = \boldsymbol{D}(m, n) = \sum_{i=1}^{N} \min(u_{mi}, u_{ni}); \qquad m, n = 1, \ldots, D. \tag{11}$$

Definition 7: An *organizational hierarchy tree (OHT)* is a directed single-root tree spanning the nodes of DCN.

Definition 8: Indirect additional coordination (or *coordination overhead*) of a DM is the amount of information flow through this node in the undirected tree of OHT under the conservation of flow constraint(s). It is found by adding the amount of coordination between all pairs of coordinating nodes with the corresponding spanning paths "passing though" a DM of interest

$$\boldsymbol{A}(m) = \sum_{i=1}^{D} \sum_{j=i+1}^{D} c_{ij} \cdot \mathbf{1}(m \in \text{ path from } i \text{ to } j)$$
$$m = 1, \ldots, D. \tag{12}$$

Definition 9: The *external organizational workload* of a DM is the sum of its external coordination and indirect additional coordination workloads

$$\boldsymbol{EW}(m) = \boldsymbol{E}(m) + \boldsymbol{A}(m); \qquad m = 1, \ldots, D. \tag{13}$$

Definition 10: The *internal organizational workload* of a DM is equal to its internal coordination

$$\boldsymbol{IW}(m) = \boldsymbol{I}(m); \qquad m = 1, \ldots, D. \tag{14}$$

Definition 11: The *organizational workload* of a DM is a weighted sum of its internal and external workload

$$\boldsymbol{W}(m) = W^I \cdot \boldsymbol{IW}(m) + W^E \cdot \boldsymbol{EW}(m); \qquad m = 1, \ldots, D. \tag{15}$$

It follows from the definitions that the overall indirect additional coordination (coordination overhead) in the organization is equal to

$$\begin{aligned}\boldsymbol{C}(OHT) &= \sum_{m=1}^{D} \boldsymbol{A}(m) \\ &= \sum_{i=1}^{D} \sum_{j=i+1}^{D} c_{ij} \\ &\quad \cdot (\{\text{number of edges between } i \text{ and } j \text{ in } OHT\} - 1).\end{aligned}$$

Example (continued): For the optimal DM-platform allocation and the corresponding DCN network shown in Fig. 3, the corresponding organizational hierarchy and workload parameters are shown in Fig. 4. In this case, the coordination overhead is $\boldsymbol{C}(OHT) = 1$.

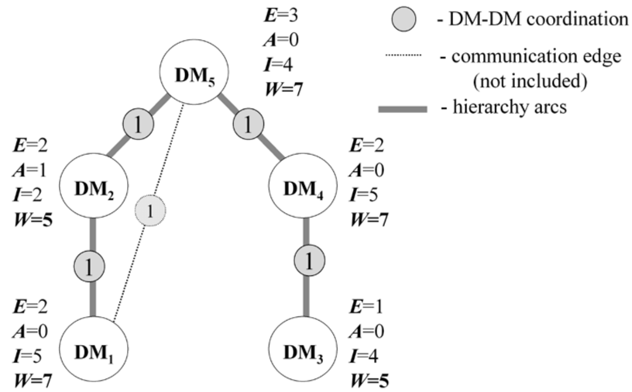

Fig. 4. Organizational hierarchy for optimal DM-platform allocation.

B. Three Objectives

In the following, we present algorithms for the Phase III to optimize three different objectives:

1) minimization of overall additional coordination imposed by the tree structure in the OHT (*minimum coordination cost* problem);
2) minimization of the maximal DM workload (*min–max* problem);
3) maximization of the aggregated coordination from the coordination links included in the OHT (*max-in* problem).

Optimization for each of these objectives produces different results. Performance comparisons among the constructed organizations over missions with different structures would validate a particular choice of the objective function and the algorithms to satisfy specific operational objectives.

C. Minimum Coordination Cost Problem

When minimizing the overall additional coordination, we use the optimal polynomial-time algorithm due to Hu [17]. The idea is to minimize the cost of coordination in OHT. We assume that the cost of a unit of coordination between any two nodes in OHT is equal to the number of nodes on the path between them. Therefore, minimization of the overall coordination cost defined in this fashion is equivalent to minimization of the additional indirect coordination. The algorithm [17] constructs Gomory–Hu tree (also called *the optimal communication tree*), which is proved to be the optimal solution for this problem.

The following definitions are used throughout the algorithm.

Definition 12: A *clique* is a set of two or more nodes of the original DCN network.

Definition 13: At any given step, all nodes of the original DCN network are partitioned into a set of cliques and of individual nodes, with a tree structure T defined over the elements of this set. Such a set is called a *partition*.

Definition 14: Residual network is a network consisting of the elements of a Partition and derived from DCN by *expanding* and/or *condensing* operations on the nodes of the current tree T. When a clique is selected in T, a residual network is obtained by condensing the components of T, which stay connected when selected clique is removed from T, and expanding the selected clique, as described in definitions 15 and 16.

Definition 15: Expanding a clique is equivalent to transforming the original DCN partition by removing this clique and by adding all nodes that constituted the clique as the individual elements of the partition.

Definition 16: A set of nodes is *condensed* when it is combined into a single node called *aggregated node*. A weight of the edge between this new *aggregated node* and any other node N in the network is equal to the sum of edge weights in the original network between N and all nodes in this aggregated node. When two cliques are condensed, it is equivalent to condensing the set of original network nodes contained in these cliques. That is, if two cliques $G_1 = \{i_1, \ldots, i_k\}$ and $G_2 = \{j_1, \ldots, j_m\}$ are to be *condensed*, the new node is $G_1 = \{i_1, \ldots, i_k, j_1, \ldots, j_m\}$ and for any node N from the original network, the edge in the residual network is

$$c_{NG}^{new} = \sum_{v=1}^{k} c_{Ni_v} + \sum_{u=1}^{m} c_{Nj_u}. \tag{16}$$

The new node is also a clique. The edge weight in the residual network between two cliques G_1 and G_2 is

$$c_{G_1G_2}^{new} = \sum_{v=1}^{k} \sum_{u=1}^{m} c_{i_v j_u}. \tag{17}$$

The algorithm is based on the following theorem [17].

Theorem: The communication cost of the tree T for a network with a set of communication requirements $\{c_{ij}, 1 \le i < j \le D\}$ is equal to the sum of cut capacities of the $(D-1)$ noncrossing cuts of this network (with cuts obtained from subsets separated by each tree edge).

A tree has one-to-one correspondence with the set of noncrossing cuts (cuts determined sequentially by removing an edge of the spanning tree and considering sets of nodes separated by the spanning tree). Since the sum of cut capacities of the $(D-1)$ noncrossing cuts represented by the Gomory–Hu tree is the least among the sums of the cut capacities of any $(D-1)$ noncrossing cuts, the theorem proves that Gomory–Hu tree is the minimum coordination cost tree.

The minimum coordination cost (*minC-cost*) algorithm is as follows.

Initialization: Start with $|T| = 1$, a tree T containing a single *clique* which consists of all nodes of the DCN.

Step 1. Select a clique G in T (which consists of more than one node of DCN). Disconnect this clique in T (remove all edges incident to this clique in T), which breaks it into several connected components. If all cliques of T contain only single nodes of the original network, STOP.

Step 2. Create a residual network by condensing each connected component into one clique (node) and expanding the selected clique.

Step 3. Pick any two nodes i and j (nodes of DCN) from the selected clique and find the minimum cut $(X, \overline{X})$ in the residual network, $i \in X, j \in \overline{X}$ (X and $\overline{X}$ consist of condensed cliques of T and of nodes of the original network from clique G).

Step 4. Create two new cliques G_1, G_2 in tree T replacing the selected clique with them:

$$G_1 = G \cap X, \qquad G_2 = G \cap \overline{X}.$$

For each clique $N \in T$ previously connected to G in T:

a) if $N \in X$, then create an edge between N and G_1

b) if $N \in \overline{X}$, then create an edge between N and G_2.

The edges are updated via

$$c_{NG_i} \leftarrow \sum_{u \in N} \sum_{v \in G_i} c_{uv}.$$

Example (continued): For the optimal DM-platform allocation and the corresponding DCN network shown in Fig. 3, the Gomory–Hu algorithm obtains an OHT shown in Fig. 4. Fig. 5 shows a step-by-step hierarchy construction process. The details of iteration (c) of the algorithm are depicted in Fig. 6.

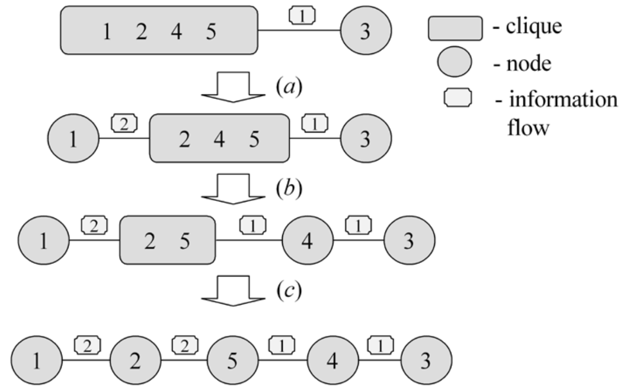

Fig. 5. Organizational hierarchy construction using Gomory–Hu algorithm.

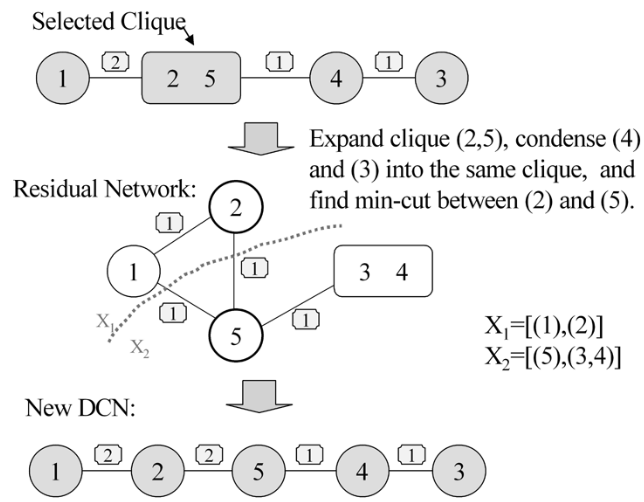

Fig. 6. Single iteration (c) of Fig. 5 in Gomory–Hu algorithm.

D. Max-In Problem

An alternative is to use maximal spanning tree algorithm to construct the OHT. We obtain a tree T that maximizes $\sum_{(i,j)\in \mathtt{E}(T)} c_{ij}$, where $\mathtt{E}(T)$ denotes the set of edges of the tree T. This can be done by applying the modified minimum spanning tree algorithm. Maximum spanning tree problem with edge weights c_{ij} transforms into a minimum spanning tree problem with edge weights $b_{ij} = c_{\max} - c_{ij}$, where $c_{\max} = \max\{c_{ij}\}$. Methods for finding the minimal spanning tree include those due to Kruskal, Jarnik–Prim–Dijkstra, and Bor'uvka (see [3], [17], and [47]).

The *max-in* algorithm is as follows.

Initialization: $T = \varnothing$

Step 1. Select an edge with maximum coordination that does not create cycles in the network.

Step 2. If ties occur, select the coordination link connected to the DM with minimal coordination workload (***CW***).

Step 3. When the number of edges in the tree is equal to D (number of DM nodes), STOP. Otherwise, go to Step 1.

The idea behind the algorithm is to include the largest coordination links and to make DMs with the largest workload to be at the lowest level of the hierarchy tree, thus obtaining a tree with maximal channel utilization.

Example (continued): Since the coordination workload of DMs is

$$[\boldsymbol{CW}(1), \boldsymbol{CW}(2), \boldsymbol{CW}(3), \boldsymbol{CW}(4), \boldsymbol{CW}(5)] = [7,\ 4,\ 5,\ 7,\ 7]$$

and DCN arc weights are the same (=1), arcs are included in the tree in the following sequence:

$$(2,\ 1),\ (2,\ 5),\ (3,\ 4),\ (4,\ 5).$$

Therefore, *max-in* algorithm produces the same result as the Gomory–Hu method for the optimal DM-platform assignment (Figs. 2 and 3).

E. Min–Max Problem

When the objective is to minimize the maximal DM workload, we impose additional constraints on the information flow. We restrict the indirect coordination to go through only one intermediate DM. If the information can be distorted while in transit, and additional intermediate nodes on the information path would increase the decision delay, it is practical to consider restrictions, such as having a single intermediate DM, to make organizations more responsive and to maximize the speed of command.

In the problem formulation, we introduce the dummy node "0" that would serve as a single-link root node. After the optimization is done, it is deleted from the tree while maintaining the tree structure.

The following variables are used to formulate the problem:

$$\begin{aligned} e_{ij} &= \begin{cases} 1, & \text{if there is a direct link from } i \text{ to } j \\ & \quad \text{in the tree} \\ 0, & \text{otherwise} \end{cases} \\ z_{ijk} &= \begin{cases} 1, & \text{if } i \text{ and } j \text{ are connected through } k \\ 0, & \text{otherwise} \end{cases} \\ l_k &= \text{level of node } k \\ W_{\text{MAX}} &= \text{maximal DM hierarchy workload.} \end{aligned}$$

The fact that we would use"direct" links accounts for the need to structure the hierarchy level by level. Then, direct links exist only from the higher level to the next lower level. The level structure of the hierarchy would be changed afterwards to place the specifically chosen DM at the root of the tree.

The following parameters are used (from the output of Phase II):

$$d_{mn} = \begin{cases} 1, & \text{if } c_{mn} > 0 \\ 0, & \text{otherwise.} \end{cases}$$

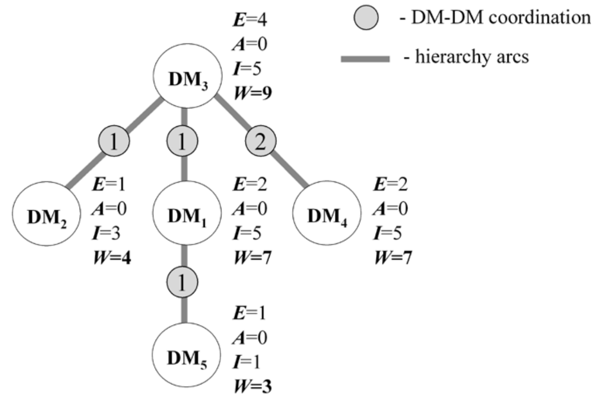

Fig. 7. Organizational hierarchy for *min-dissimilarity* DM-platform clustering.

Following [29], the problem assumes the form of a binary (0–1) programming problem

$$\min W_{\text{MAX}}$$
$$\begin{cases} \sum_{i,\,j=1}^{D} e_{ij} = D - 1 \\ \sum_{j=0}^{D} e_{j0} = 0,\ \sum_{j=0}^{D} e_{ji} = 1, \qquad i = 1, \ldots, D \\ l_j \geq l_i + 1 + (e_{ij} - 1)(D+1), \qquad i,\, j = 0, \ldots, D \\ e_{ij} + e_{ji} + \sum_{k=1}^{D} z_{ijk} \geq d_{ij}, \qquad i,\, j = 1, \ldots, D \\ e_{ik} + e_{ki} + e_{jk} + e_{kj} \geq 2z_{ijk}, \qquad i,\, j,\, k = 1, \ldots, D \\ W_{\text{MAX}} \geq W^I \cdot I(n) + W^E \\ \quad \cdot \left(E(n) + \sum_{i<j} z_{ijn} c_{ij} \right), \qquad n = 1, \ldots, D \\ e_{ij},\, z_{ijk} \in \{0,\, 1\} \end{cases} \qquad (18)$$

After a solution to this problem is found, the"dummy" root node is discarded. Then the node with some specific property (e.g., minimum hierarchy depth, maximum workload, maximum coordination) is found and selected to be at the root of the organizational hierarchy.

Example (continued): Figs. 4, 7, and 8 show organizational structures corresponding to these coordination networks. These structures are obtained by minimizing the maximal DM workload (*min–max* problem) via the solution of 0–1 binary programming problems. Here, optimizing the other two objectives (*minC-cost* and *max-in* problems) produces identical results in each case. The organization corresponding to optimal clustering has the least maximal DM workload.

Experiments suggest that strong dependency exists between the density of coordination networks and the performance of the corresponding organizational structures. Density of

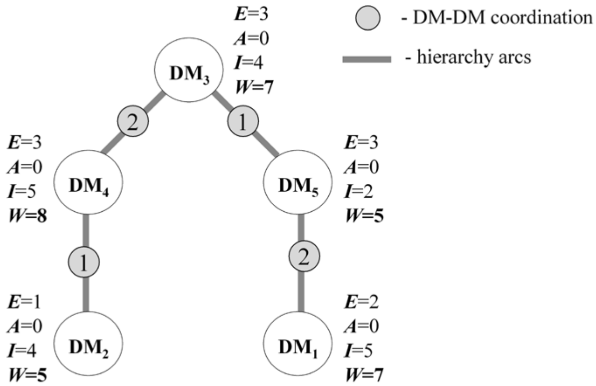

Fig. 8. Organizational hierarchy for *max-similarity* DM-platform clustering.

TABLE I
AVERAGE MAXIMAL COORDINATION WORKLOAD (CW) FOR PHASE III

number of DMs	Min Dissimilarit	Max Similarity	Best Merge	Optimal Algorithm
2	51.69	51.76	51.17	50.74
3	42.2	41.82	41.62	41.1
4	33.52	33.06	33.18	32.88
5	33.1	32.42	31.95	31.87
6	33.7	31.33	30.13	30.13
7	24.26	24.26	24.26	24.26

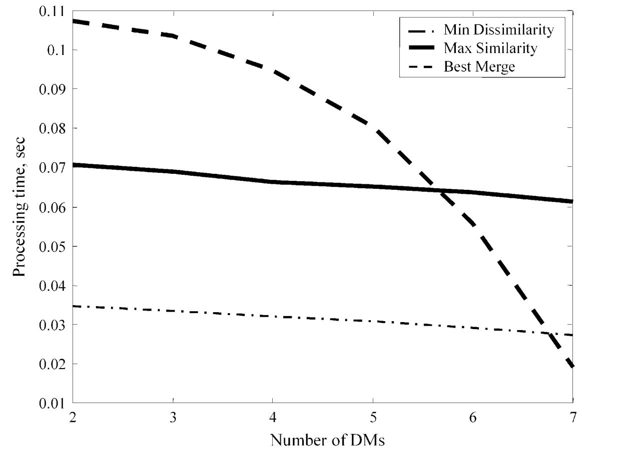

Fig. 9. Average CPU time of heuristic algorithms. Number of platforms = 7; number of simulations = 100.

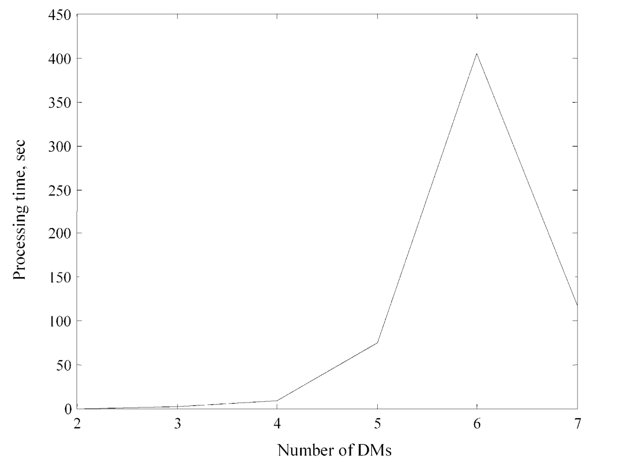

Fig. 10. Average CPU time of optimal algorithm. Number of platforms = 7; number of simulations = 100.

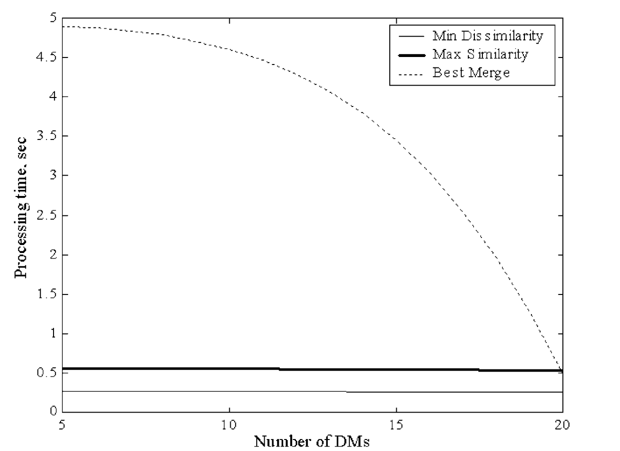

Fig. 11. Average CPU time of heuristic clustering algorithms. Number of platforms = 20; number of simulations = 500.

coordination networks is dependent on the choice of workload weights (internal–external workload tradeoff). The following section discusses results from an experiment with the DDD-III simulator obtained by varying workload weight parameters.

V. SIMULATION RESULTS

A. Algorithm Performance: DM-Resource Allocation

Simulation results for the clustering algorithms (based on the scheduling results obtained by the MDLS method with critical path task selection) are shown in Table I and Figs. 9–11 (workload weights are $W^I = 10$, $W^E = 1$; see Appendix for information on random problem generation). For a problem with 30 tasks and seven platforms, Table I shows the average maximal coordination workload (objective function) of clustering algorithms. The average CPU times (for Pentium 600 MHz processor) of heuristic algorithms and of the optimal procedure are presented in Figs. 9 and 10, respectively. Results are based on 100 Monte Carlo simulations. Fig. 11 shows similar CPU time behavior for a problem with 30 tasks and 20 platforms for heuristic algorithms only (the optimal algorithm is computationally infeasible for large-size problems).

For heuristic algorithms of Phase II, the processing time of hierarchical clustering procedure increases as the potential number of DMs (number of clusters) is reduced. The comparison of *max-similarity* and *min-dissimilarity* algorithms shows that the *min-dissimilarity* algorithm has the least processing time, while *max-similarity* method produces on average better results. On the other hand, these algorithms are significantly faster than best-merge procedure, although the latter is closer to the optimal solution. The optimal solution has exponential complexity and can only be used for small-size problems (see Fig. 10).

B. Algorithm Performance: Organization Structure

In this section, we discuss the performance of algorithms to obtain DM–DM coordination hierarchy. Each of these algo-

TABLE II
AVERAGE CPU TIMES OF ALGORITHMS FOR PHASE III (IN SECONDS)

number of DMs	Min C-Cost	Max-In	Min-Max
2	0.007	0.0001	0.0039
3	0.0309	0.0001	0.0089
4	0.0897	0.0001	0.0344
5	0.193	0.0004	0.4669
6	0.351	0.0011	5.222
7	0.526	0.0012	105.8115

rithms finds an optimal solution to the corresponding problem. The goal of such simulations is to select the number of DMs for an organization. We try to achieve a tradeoff between optimizing the average and the maximal workloads of DMs in an organization. It would provide us with a balanced workload among members of the organization (better resource utilization), as well as prevent individual DMs from being overloaded. On the other hand, having under-loaded DMs provides us with redundancies, making the organization more robust to failures.

As indicated earlier, the complexity of Gomory–Hu algorithm (*minC-cost* problem) and maximum spanning tree algorithm (*max-in* problem) is polynomial, while *min–max* algorithm has exponential complexity (over the number of DMs, tasks, and platforms). This is illustrated by the average CPU-time data for each algorithm shown in Table II. Fig. 12 shows the average maximal DMs workload for each algorithm. Fig. 13 presents the average coordination, for which maximum spanning tree algorithm is optimal. Fig. 14 illustrates the average coordination overhead. Gomory–Hu algorithm is optimal for this case. As indicated in Fig. 15, Gomory–Hu algorithm obtains organizations with the best average DM workload.

From simulation results, we conclude that the min–max algorithm obtains a solution with the least maximal DM workload (when the number of DMs is small) by distributing the workload among DMs. However, the complexity of this algorithm prohibits its use for large-size problems. Furthermore, we note that, as the number of DMs increases, the constraints on coordination, placed by min–max problem formulation, make its solution nonoptimal. max spanning tree method produces solutions with better maximal workload for larger number of DMs at the expense of increased coordination overhead and average DM workload. It is also significantly faster than Gomory–Hu algorithm.

From the maximal and average workload data [see Figs. 12 and 15] we conclude that the choice of four or five DMs is the best for this problem (workload weights are $W^I = 10$, $W^E = 1$).

C. Effects of Internal/External Workload Weights on Organizational Structure

In this section, we explore the behavior of organizational structures obtained via Gomory–Hu (*minC-cost*) algorithm. The results are based on clustering data obtained from *min-dissimilarity* clustering algorithm.

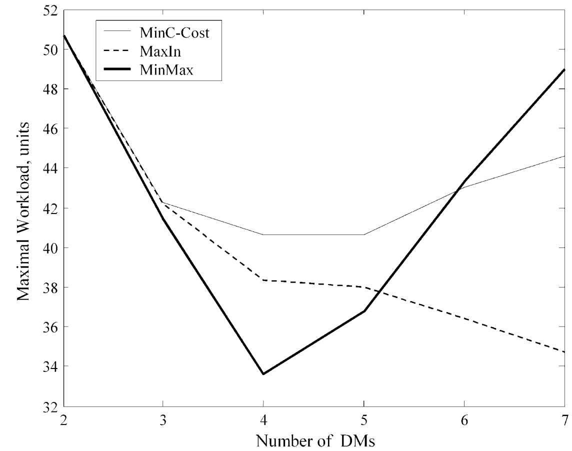

Fig. 12. Maximal workload of DM. Number of platforms = 7; number of simulations = 100.

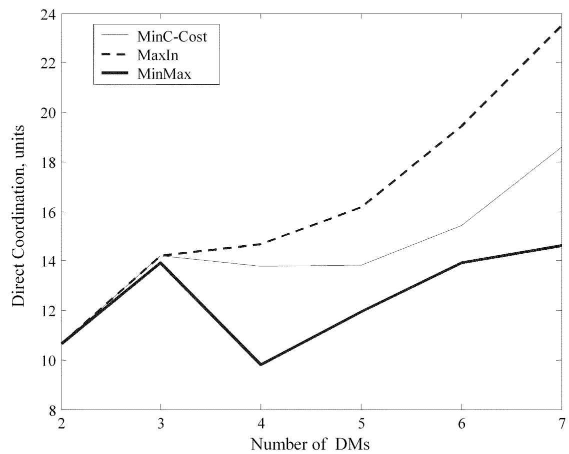

Fig. 13. Average direct coordination. Number of platforms = 7; number of simulations = 100.

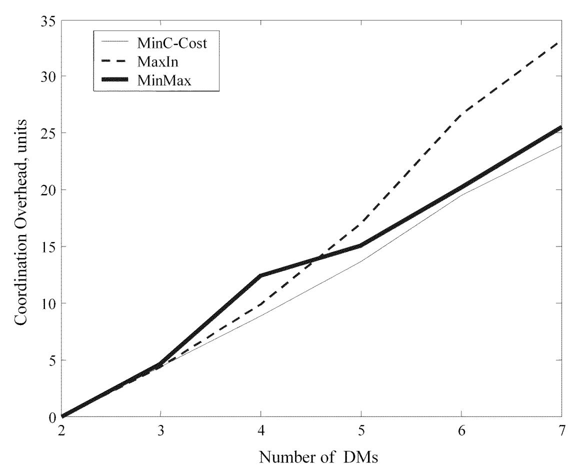

Fig. 14. Average external coordination overhead. Number of platforms = 7; number of simulations = 100.

The shape of organizational structures for a 5-node organization with min-depth root selection for our example varies

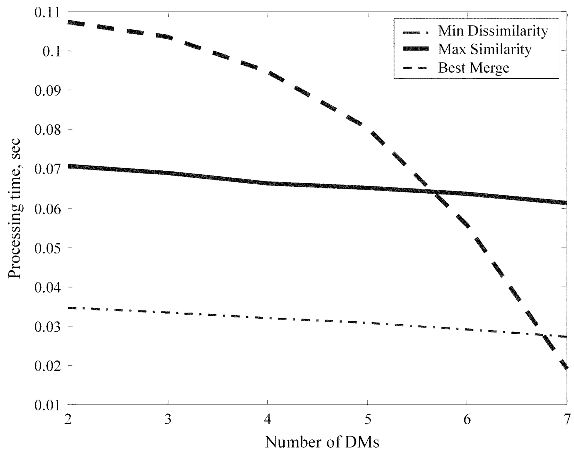

Fig. 15. Average CPU time of heuristic algorithms. Number of platforms = 7; number of simulations = 100.

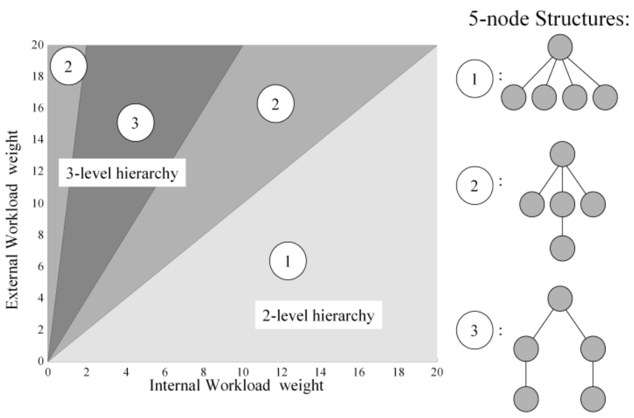

Fig. 16. Effects of workload weights on 5-node organizational structure.

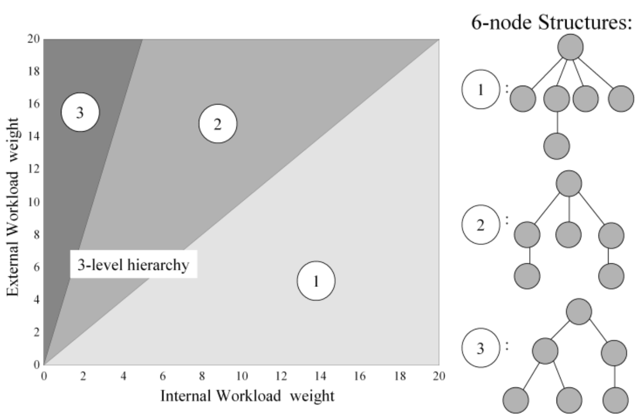

Fig. 17. Effects of workload weights on 6-node organizational structure.

between three basic hierarchies according to the internal/external workload emphasis (see Fig. 16). Fig. 17 shows similar results for 6-node organizations. Although there are five possible 6-node architectures, organizational hierarchies obtained using our design process vary among three distinct 3-level structures only. Each shaded area in Figs. 16 and 17 corresponds to the hierarchy structure (obtained by Gomory–Hu algorithm for a pair of internal and external workload weights in this area) shown on the right-hand side of the figure.

Minimizing the internal DM workload in Phase II (high ratio of internal workload weight to external weight) results in DM-resource allocation with heavy inter-DM coordination. In our examples, we obtained dense coordination networks with evenly spread coordination. Optimization of structures for dense and evenly distributed coordination networks leads to flat hierarchy structures (low-level hierarchies). This comes from the fact that a flat hierarchy minimizes the communication cost $\boldsymbol{C}(OHT)$ (coordination overhead) of OHT for coordination networks with such properties. On the other hand, optimization for sparse coordination networks results in multilayered organizational hierarchies.

VI. Summary and Future Research

Different organizations exhibit differences in their performance. Even for small organizations facing missions with a limited number of tasks, there can be an enormous number of possible solutions to the organizational design problem. Optimization can yield significant improvements in performance. In this paper, we presented Phases II and III of our 3-phase process for optimizing the organizational design (outlined in [56]). We provided mathematical formulations of DM-resource allocation (Phase II) and coordination structure optimization (Phase III) problems, and presented algorithms to solve these problems. We have also shown the dependence between the applied optimization criteria and the structural behavior of organizations obtained via our design process.

Our current efforts are focused on conducting a comparative analysis of various optimization techniques in *solving specific design problems* and on defining criteria for classifying multiobjective optimization problems into groups that require different optimization strategies to *reduce solution complexity* for large-scale design problems. We also look to define measures of *organizational robustness* (i.e., the ability of an organization to maintain the required level of performance despite variations in its mission environment) and of *adaptability* (i.e., the *ability of an organization to adapt* to environmental changes and functional failures). Developing fast algorithms for real-time analysis of feasible adaptation options to suggest *suitable* forms of adaptation and appropriate *transition sequence* for reconfiguration would provide a computational framework for on-line adaptation in complex C2 systems facing uncertain and volatile environments.

Appendix
Random Problem Generation

Resources: Resource length varies uniformly between four and ten. Elements of requirement/capability vectors vary uniformly between one and five units.

Tasks: Number of tasks is fixed. Task positions are uniform in [0, 50] $\times$ [0, 50] grid.

Task Precedence Graph: Number of levels is uniformly distributed according to task-to-level ratio. Max and min task-to-level ratios are, respectively, 0.6 and 0.25. Number of predecessors of each task from upper levels (>1) is 2.

Max number of tasks per level is six tasks. Maximum task processing time is 50 units.

Platforms: Number of platforms is fixed. Platform velocity varies between one and three units.

REFERENCES

[1] S. Androile, *Storyboard Prototyping*. Wellesley, MA: QED Information Sciences, 1991.

[2] L. Argote and D. Epple, "Learning curves in manufacturing," *Science*, vol. 247, 1990.

[3] D. P. Bertsekas, *Network Optimization: Continuous and Discrete Models*. New York: Athena Scientific, 1998.

[4] K. S. Cameron, "Effectiveness as paradox: Consensus and conflict in conceptions of organizational effectiveness," *Manage. Sci.*, vol. 32, no. 5, pp. 539–553, 1986.

[5] K. M. Carley and M. J. Prietula, "Computational organization theory,", Hillsdale, NJ, 1994.

[6] J. C. Duchon and T. J. Smith, "Extended workdays and safety," *Int. J. Ind. Ergon.*, vol. 11, pp. 37–49, 1993.

[7] J. C. Dyer, "Team research and team training: State-of-the-art review," in *Human Factors Review*, F. A. Muckler, Ed. Santa Monica, CA: Human Factors Society, 1984, pp. 285–323.

[8] T. Elbert, W. Ray, Z. Kowalik, S. Skinner, K. Graf, and N. Birbaumer, "Chaos and physiology," *Phys. Rev.*, vol. 74, pp. 1–47, 1994.

[9] E. E. Entin, "Optimized command and control architectures for improved process and performance," in *Proceedings of the 1999 Command & Control Research & Technology Symp*. Newport, RI: Naval War College, 1999.

[10] E. E. Entin and D. Serfaty, "Adaptive team coordination," *J. Hum. Factors*, vol. 41, no. 2, pp. 321–325, 1999.

[11] B. S. Everitt, *Cluster Analysis*. New York: Wiley, 1993.

[12] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA: Freeman, 1979.

[13] H. Guetzkow and H. Simon, "The impact of certain communication nets upon organization and performance in task-oriented groups," *Manage. Sci.*, vol. 1, pp. 233–250, 1955.

[14] T. C. Hankins and G. F. Wilson, "A comparison of heart rate, eye activity, EEG and subjective measures of pilot mental workload during flight," *Aviation, Space, and Environmental Medicine*, vol. 69, no. 4, pp. 360–367, 1998.

[15] S. G. Hart and L. E. Staveland, "Development of NASA-TLX (task load index): Results of experimental and theoretical research," in *Human Mental Workload*, P. A. Hancock and N. Meshkati, Eds. Amsterdam, The Netherlands: North Holland, 1988, pp. 139–183.

[16] E. Houstis, E. Gallopoulos, R. Bramley, and J. Rice, "Problem-solving environments for computational science," *IEEE Comput. Sci. Eng. Mag.*, July–Sept. 1997.

[17] T. C. Hu, *Combinatorial Algorithms*. Reading, MA: Addison-Wesley, 1982.

[18] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, ser. Prentice-Hall Advanced Reference Series: Computer Science. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[19] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrica*, vol. 32, pp. 241–254.

[20] D. Kleinman, K. Pattipati, P. Luh, and D. Serfaty, "Mathematical models of team performance: A distributed decision-making approach," in *Teams: Their Training and Performance*, R. Swezey and E. Salas, Eds. Norwood, NJ: Ablex, 1992, pp. 177–218.

[21] D. L. Kleinman and D. Serfaty, "Team performance assessment in distributed decision-making," in *Proc. Symp. Interactive Networked Simulation for Training*, Orlando, FL, 1989.

[22] G. N. Lance and W. T. Williams, "A general theory of classificatory sorting strategies—Part II: Clustering systems," *Comput. J.*, vol. 10, pp. 271–277, 1967.

[23] P. R. Lawrence and J. W. Lorsch, "Organization and environment: Managing differentiation and integration," Graduate School of Business Administration, Boston, MA, 1967.

[24] Y. N. Levchuk, K. R. Pattipati, M. L. Curry, and M. Shakeri, "Design of congruent organizational structures: Theory and algorithms," in *Proc. Command and Control Research and Technology Symp.*, Monterey, CA, June 1996.

[25] Y. N. Levchuk, K. R. Pattipati, and M. L. Curry, "Normative design of organizations to solve a complex mission: Theory and algorithms," in *Proc. Command and Control Research and Technology Symp.*, Washington, DC, June 1997.

[26] Y. N. Levchuk, K. R. Pattipati, and D. L. Kleinman, "Designing adaptive organizations to process a complex mission: Algorithms and applications," in *Proceedings of the 1998 Command & Control Research & Technology Symposium*. Monterey, CA: Naval Postgraduate School, June 1998.

[27] ——, "Analytic model driven organizational design and experimentation in adaptive command and control," *Syst. Eng.*, vol. 2, no. 2, 1999.

[28] Y. N. Levchuk, J. Luo, G. M. Levchuk, K. R. Pattipati, and D. L. Kleinman, "A multifunctional software environment for modeling complex missions and devising adaptive organizations," in *Proceedings of the 1999 Command & Control Research & Technology Symposium*. Newport, RI: Naval Postgraduate School, June 1999.

[29] G. M. Levchuk, Y. N. Levchuk, J. Luo, F. Tu, and K. R. Pattipati, "A library of optimization algorithms for organizational design," in *Proceedings of the 2000 Command & Control Research & Technology Symposium*. Monterey, CA: Naval Postgraduate School, June 2000.

[30] Y. Levchuk, C. Pattipati, and D. Kleinman, "Designing adaptive organizations to process a complex mission: Algorithms and applications," in *Proc. Command and Control Research and Technology Symp.*. Monterey, CA, 1998, pp. 11–32.

[31] Z. Linn and K. M. Carley, "Maydays and Murphies: A study of the effects of organizational design, task, and stress on organizational performance," in *Annu. Meeting Amer. Sociol. Assoc.*, Pittsburg, PA, 1992.

[32] K. D. Mackenzie, *Organizational Design: The Organizational Audit and Analysis Technology*. Norwood, NJ: Ablex, 1986.

[33] C. F. Mackenzie, N. J. Jefferies, W. A. Hunter, W. N. Bernhard, Y. Xiao, and R. Horst, "Comparison of self reporting of deficiencies in airway management with video analysis of actual performance," *Hum. Factors*, vol. 38, no. 4, pp. 623–635, 1996.

[34] J. MacMillan and E. B. Entin, "Decision-oriented display design and evaluation," in *Proceedings of the 1991 Symposium on Command & Control Research*. Washington, DC: NDU, 1991.

[35] M. L. Metersky, J. M. Ryder, and M. A. Leonardo, "A change in system-design emphasis: From machine to human," in *Advanced Technology in C3I System Engineering*, S. Andriole, Ed. Washington, DC: AFCEA International Press, 1989.

[36] C. M. Mitchell, "Operator models, model-based displays, and intelligent aiding," in *Human/Technology Interaction in Complex Systems*, W. B. Rouse, Ed. Greenwich, CT: JAI Press, 1996, vol. 8, pp. 67–172.

[37] A. Pete, D. L. Kleinman, and K. R. Pattipati, "Structural congruence of tasks and organizations," in *Proceedings of the 1994 Symp. on Command and Control Research and Decision Aids*. Monterey, CA: Naval Postgraduate School, 1994, pp. 168–175.

[38] A. Pete, K. R. Pattipati, D. L. Kleinman, and Y. N. Levchuk, "An overview of decision networks and organizations," *IEEE Trans. Syst., Man, Cybern.*, vol. 28, pp. 172–192, May 1998.

[39] R. Ramesh and D. H. Andrews, "Distributed mission training: Teams, virtual reality, and real-time networking," *Commun. ACM*, vol. 42, no. 9, pp. 65–67, 1999.

[40] K. Roberts, "New challenges to organizational research: High reliability organizations," *Ind. Crisis Q.*, vol. 3, no. 3, pp. 111–125, 1989.

[41] D. L. Saisi and D. Serfaty, "Model-based human interface design for distributed tactical command and control," in *Proc. JDL Command and Control Research Symp.*, Monterey, CA, 1988.

[42] E. Salas, T. L. Dickinson, S. A. Converse, and S. I. Tannenbaum, *Toward and Understanding of Team Performance and Training in Teams: Their Training and Performance*, R. W. Swezey and E. Salas, Eds. Norwood, NJ: Ablex, 1992.

[43] E. Salas, J. A. Cannon-Bowers, and J. H. Johnston, "How can you turn a team of experts into an expert team?: Emerging training strategies," in *Naturalistic Decision Making*, C. Zsambok and G. Klein, Eds. Hillsdale, NJ: LEA, 1997, pp. 359–370.

[44] H. A. Simon, *The Sciences of the Artificial*. Cambridge, MA: MIT Press, 1981.

[45] K. A. Smith-Jentsch, R. L. Zeisig, B. Acton, and J. A. McPherson, "Team dimensional training: A strategy for guided team self-correction," in *Making Decisions Under Stress*, J. A. Cannon-Bowers and E. Salas, Eds. Washington, DC: American Psychological Association, 1998, pp. 271–297.

[46] P. H. A. Sneath and R. R. Sokal, *Numerical Taxonomy*. San Francisco, CA: Freeman, 1973.

[47] R. E. Tarjan, *Data Structures and Network Algorithms*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1985.

[48] D. I. Tepas, M. J. Paley, and S. P. Popkin, "Work schedules and sustained performance," in *Handbook of Human Factors & Ergonomics*, Second ed, G. Salvendy, Ed. New York: Wiley, 1997, pp. 1021–1058.

[49] A. Tziner and D. Eden, "Effects of crew composition on crew performance: Does the whole equal the sum of its parts?," *J. Appl. Psych.*, vol. 70, no. 1, pp. 85–93, 1985.

[50] P. A. Vail, E. B. Entin, J. MacMillan, and J. C. Deckert, "An experiment in decision aiding for the choice of SDS defense employment option," in *Proceedings of the 1991 Symposium on Command & Control Research*. Washington, DC: NDU, 1991.

[51] J. H. Ward, "Hierarchal grouping to optimize an objective function," *J. Amer. Stat. Assoc.*, vol. 58, pp. 236–244, 1963.

[52] Y. Xiao and C. F. Mackenzie, "Uncertainty in trauma resuscitation," in *Proc. Human Factors and Ergonomics Society 41st Annu. Meeting*, Santa Monica, CA, 1997.

[53] American Trucking Association, Inc., *Trucking Research Institute. Eye-Activity Measures of Fatigue and Napping as a Fatigue Countermeasure*. Washington, DC: American Trucking Association, 1999.

[54] Federal Aviation Administration, "Pilot fatigue, sleep and circadian rhythms: NASA fatigue countermeasures program," *Aviation Safety J.*, vol. 3, no. 1, pp. 20–25, 1993.

[55] Naval Air Warfare Center Training System Division (NAWC-TSD), Team Training Bibliography, 1998.

[56] G. M. Levchuk, Y. N. Levchuk, J. Luo, K. R. Pattipati, and D. L. Kleinman, "Normative design of organizations—Part I: Mission planning," *IEEE Trans. Syst., Man, Cybern. A*, vol. 32, pp. 346–359, May 2002.